Projet n°10 : Portail captif wifi

Table des matières

IPrésentation....2
I.1Intérêt....2
I.2Description de l'étude initial....2
I.3Fonctionnement générique d'un portail captif....3
IIAlternative au portail captif....4
II.1EAP....4
II.2802.1x....4
II.3Radius....5
II.4Conclusion....6
IIISécurité de la connexion....7
IVComparaison entre portails captifs....8
IV.1Caractéristiques principales....8
IV.2Choix....9
VTest et comparatif entre NoCat et WifiDog....10
V.1Matériel requis....10
V.2Installation et configuration succinct de NoCat....10
V.3Installation et configuration succinct de WifiDog....11
VIArchitecture logiciel....13
VI.1Analyse....13
VI.2Ajout de fonctionnalités....14
VIICahier des charges....15
VIIIPlanning prévisionnel....16

I Présentation

I.1 Intérêt

De plus en plus de personnes disposent aujourd'hui d'un organe mobile (portable, PDA, ...) et souhaitent pouvoir accéder à l'Internet avec cet organe dans la majorité des lieux qu'ils fréquentent. Dans cet optique, l'expansion très rapide des points d'accès sans-fil permet la connexion des appareils nomades. Néanmoins chaque réseau possède sa politique d'accès et ne souhaite pas laisser n'importe qui accéder aux ressources réseaux. Ainsi il est nécessaire de mettre en place des systèmes d'authentification sur ces réseaux qui doivent cumuler de multiples avantages comme une compatibilité avec la majorité des appareils mobiles du marché, sécuriser les échanges entre les client et le réseau, offrir à l'utilisateur la plus grande transparence aussi bien lors de la phase d'authentification que lors de l'utilisation du réseau, réduire l'impact au niveau des ressources matériels et de la bande passante.

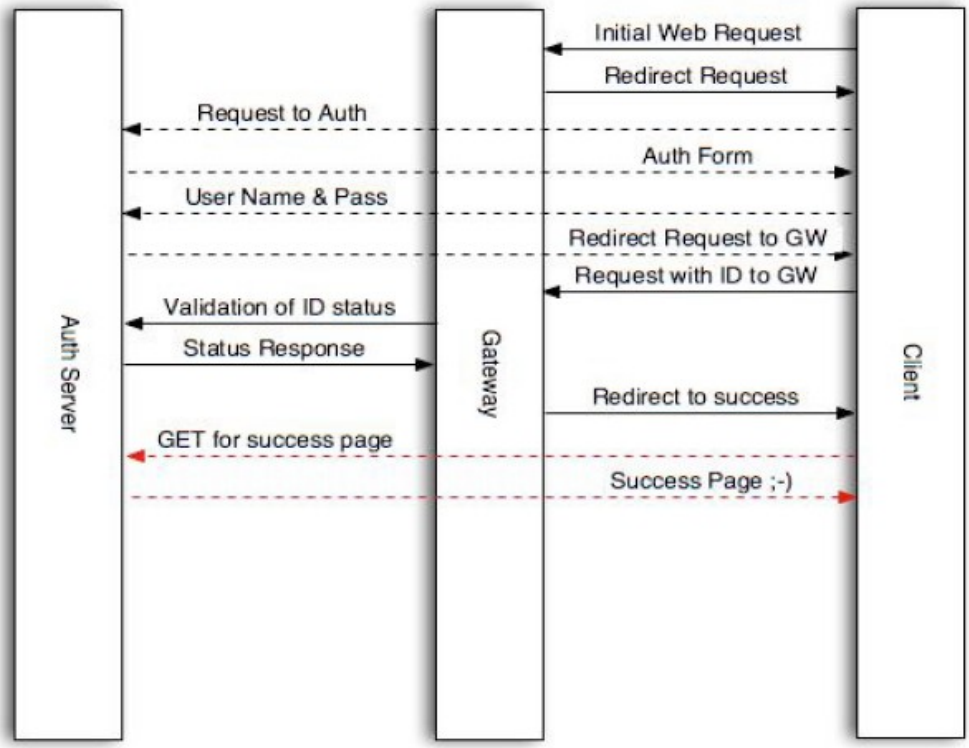
Devant ces enjeux, le portail captif s'est imposé comme une solution fréquemment utilisée dans les points d'accès payants, mais elle peut se généraliser à tous types de réseau (sans-fil ou filaire) nécessitant un contrôle d'accès. La technique du portail captif consiste à forcer un client souhaitant accéder à un service en http ou htpps sur un réseau à passer par une page web servant à l'authentifier sur ce réseau. Après authentification, la passerelle peut le laisser passer; le client a accès au réseau de façon classique.

Ainsi le système utilisé dans notre université permet d'authentifier les utilisateurs désirant se connecter depuis n'importe quelle machine apportées à l'intérieur des bâtiments; la seule nécessité est de disposer d'un compte utilisateur valide, d'un navigateur web gérant le htpps et dans le cas du portail captif NoCat utilisé actuellement du javascript.

I.2 Description de l'étude initial

Notre étude se base sur la comparaison entre plusieurs solutions de portails captifs. Nous avons ainsi comparé leur différence sur le papier en mettant face à face leurs points forts et points faibles respectifs. Nous avons ensuite étudiés plus particulièrement NoCat (solution actuellement mise en place) et WifiDog en examinant l'installation, les ressources nécessaires, leur procédure d'authentification, leur sécurité. Ce dernier point nous a particulièrement interpellé et nous avons examiné d'autres possibilités des réseaux sans fil permettant de sécuriser au maximum les échanges entre les clients et le point d'accès sans-fil. Grâce à ce large éventail de solutions possibles, nous avons pu établir un cahier des charges du portail captif offrant les meilleures fonctionnalités.

I.3 Fonctionnement générique d'un portail captif



Authentification (cas de WifiDog)

Le client se connecte au réseau par l'intermédiaire d'un connexion filaire ou au point d'accès pour du wifi. Ensuite un serveur dhcp lui fournit une adresse ip ainsi que les paramètres de la configuration du réseau. A ce moment-là, le client a juste accès au réseau entre-lui et la passerelle, cette dernière lui interdisant, pour l'instant, l'accès au reste du réseau. Lorsque le client va effectuer sa première requête de type web en http ou https, la passerelle le redirige vers une page web d'authentification qui lui permet de s'authentifier grâce à un login et un mot de passe. Cette page est cryptée à l'aide du protocole ssl pour sécuriser le transfert du login et du mot de passe. Le système d'authentification va alors contacter une base de données contenant la liste des utilisateurs autorisés à accéder au réseau. Enfin le système d'authentification indique, plus ou moins directement selon les portails captif, à la passerelle que le couple mac/ip du client est authentifié sur le réseau. Finalement le client est redirigé vers la page Web qu'il a demandé initialement; le réseau derrière la passerelle lui est dorénavant accessible. Le portail captif, grâce à divers mécanismes comme une fenêtre pop-up sur le client rafraîchie à intervalles réguliers ou des requêtes ping vers le client, est en mesure de savoir si l'utilisateur est toujours connecté au réseau. Au bout d'un délai d'absence sur le réseau, le portail captif va couper l'accès à cet utilisateur.

II Alternative au portail captif

Il nous a semblé intéressant d'étudier les autres mécanismes d'authentification, plus particulièrement sur les réseaux sans-fil, existants afin d'avoir le plus large panorama possible de solutions.

II.1 EAP

Le fonctionnement du protocole EAP est basé sur l'utilisation d'un contrôleur d'accès (authenticator), chargé d'établir ou non l'accès au réseau pour un utilisateur (supplicant). Le contrôleur d'accès est un simple garde-barrière servant d'intermédiaire entre l'utilisateur et un serveur d'authentification (authentication server), il ne nécessite que très peu de ressources pour fonctionner. Dans le cas d'un réseau sans fil, c'est le point d'accès qui joue le rôle de contrôleur d'accès.

Le serveur d'authentification permet de valider l'identité de l'utilisateur, transmis par le contrôleur réseau, et de lui renvoyer les droits associés en fonction des informations d identification fournies. De plus, un tel serveur permet de stocker et de comptabiliser des informations concernant les utilisateurs afin, par exemple, de pouvoir les facturer à la durée ou au volume (util d un point de vue commerciale).

La plupart du temps le serveur d'authentification est un serveur RADIUS (Remote Authentication Dial In User Service), un serveur d'authentification standard défini par les RFC 2865 et 2866, mais tout autre service d'authentification peut être utilisé.

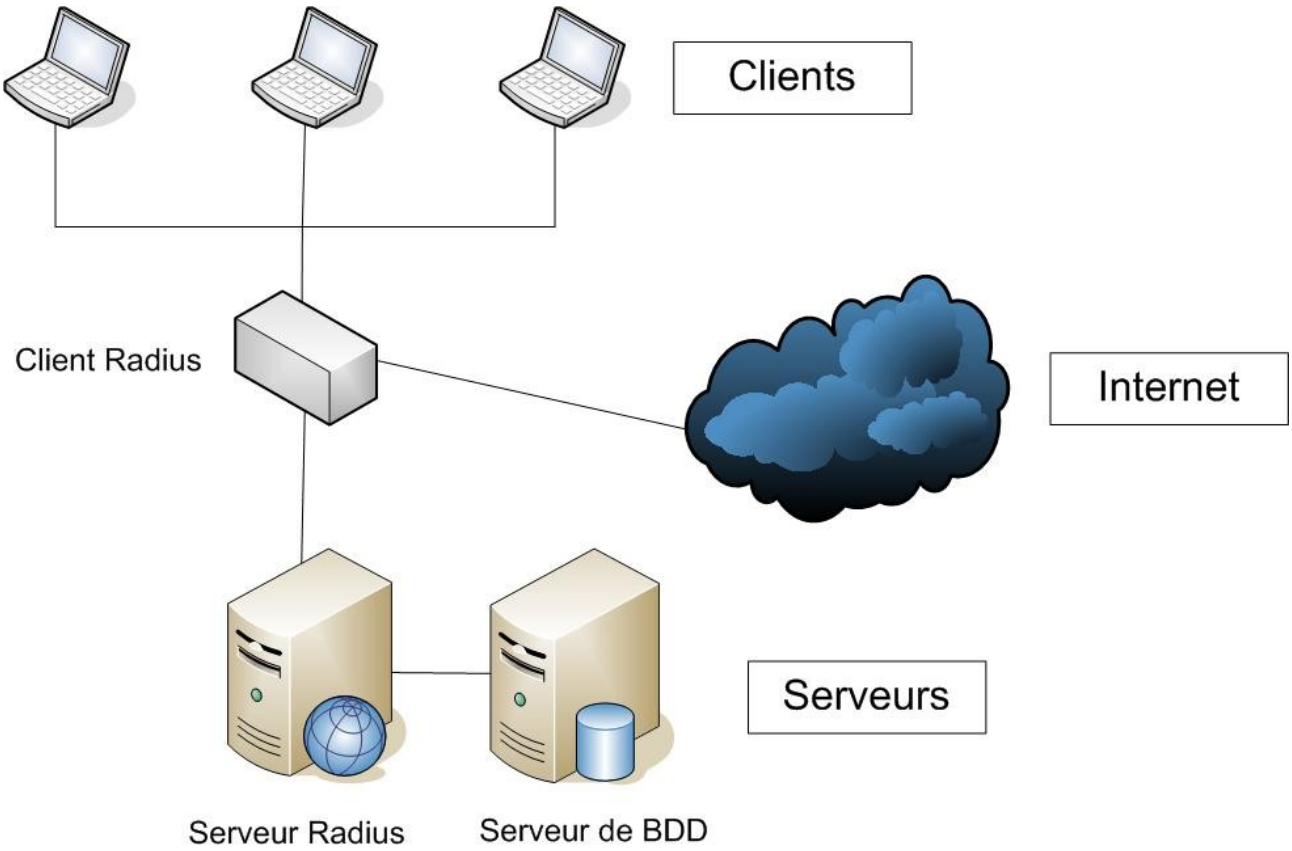
II.2 802.1x

Le standard 802.1x est une solution de sécurisation, mise au point par l'IEEE en juin 2001, permettant d'authentifier un utilisateur souhaitant accéder à un réseau grâce à un serveur d authentification.

Le 802.1x repose sur le protocole EAP (Extensible Authentication Protocol), défini par l'IETF, dont le rôle est de transporter les informations d'identification des utilisateurs.

802.1x est un protocole assurant l'identification par port pour l'accès à un réseau; il n'est pas lié explicitement à RADIUS dans son principe et utilise dans toutes ses définitions les termes "serveur AAA" et "protocole AAA", mais il se trouve que l'annexe D du document de référence mentionne comme "exemple" de protocole et serveur AAA le seul RADIUS. Toutes les mises en S uvre de 802.1X connues s'appuient donc sur RADIUS.

II.3 Radius



Radius est un protocole d'authentification à distance mis au point par la société Livingston Enterprises et conçu pour gérer les connexions d'utilisateurs à des services distants, notamment ce qui relève de la politique d'autorisation, des droits d'accès et de la traçabilité (notamment pour maintenir un état précis des connexions et une "comptabilité" des connexions). Ce processus d'authentification est entièrement crypté et doit être relié à une source d'informations, souvent un annuaire LDAP.

Le fonctionnement de RADIUS est basé sur un système client/serveur chargé de définir les accès d'utilisateurs distants à un réseau. Il s'agit du protocole de prédilection des fournisseurs d'accès à Internet car il est relativement standard et propose des fonctionnalités de comptabilité permettant aux FAI de facturer précisément leurs clients.

Le protocole RADIUS repose principalement sur un serveur (le serveur RADIUS), relié à une base d'identification (base de données, annuaire LDAP, etc.) et un client RADIUS, appelé NAS (Network Access Server), faisant office d'intermédiaire entre l'utilisateur final et le serveur. L'ensemble des transactions entre le client RADIUS et le serveur RADIUS est chiffrée et authentifiée grâce à un secret partagé.

L'authentification s effectue généralement sur le port 1645 ou 1812 alors que l accouting (journalisation des accès et facturation) s effectue sur le port 1646 ou 1813. Les communications client-serveur ont lieu en UDP.

Remarque: RADIUS assure un transport en clair, seul le mot de passe est chiffré par hachage ; la sécurité toute relative du protocole repose sur le seul shared secret et impose la sécurisation des échanges entre le client et le serveur par sécurité physique ou VPN. RADIUS n'assure pas de mécanisme d'identification du serveur ; or se faire passer pour un serveur est un excellent moyen de récolter des noms et mots de passe.

II.4 Conclusion

Le 802.1x constitue une solution intéressante, cependant elle oblige à posséder des points d'accès gérant ce protocole et la connexion à un serveur Radius. On se retrouve ainsi vite limité afin de pouvoir apporter des améliorations à ce système, il n'est pas propice à un développement avec un cahier des charges spécifiques. Concernant Radius, il présente l'intérêt de présenter une interface unique de gestion des authentifications en faisant abstraction de diverses base de données en aval. Néanmoins cet intérêt est peu utile dans notre cas.

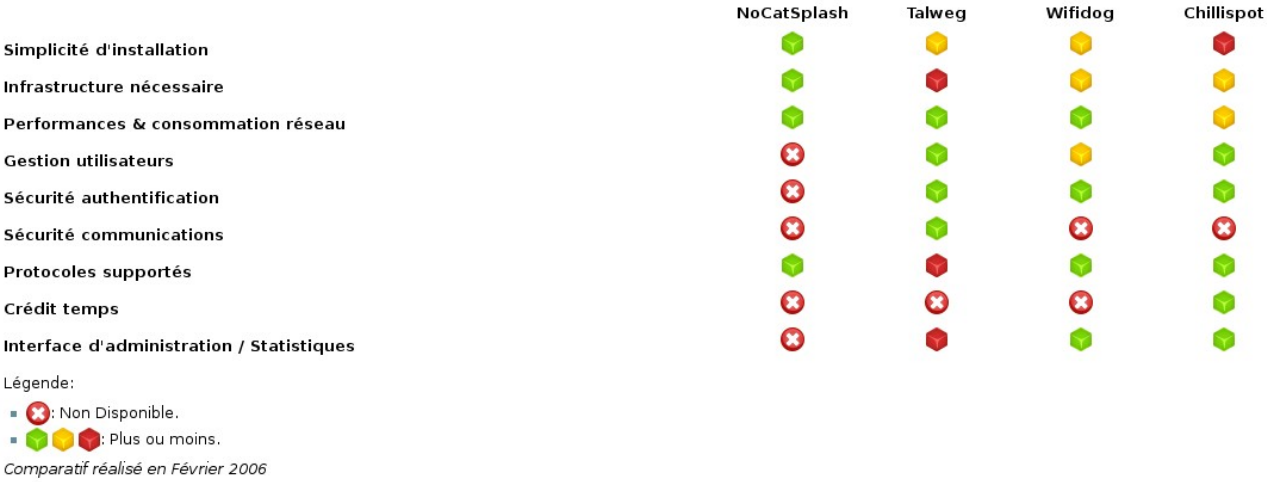
III Sécurité de la connexion

De part leur fonctionnement, les communications entre le client et le point d'accès ne sont pas chiffrés. De plus le réseau wifi agissant comme un hub, il est possible d'accéder à l'ensemble des données échangées par les clients connectés en wifi. En utilisant la technique du Man in the Middle par flood arp ciblé, on peut arriver à se faire passer pour la passerelle auprès des clients connectés au wifi. Par conséquent on peut proposer sa propre fenêtre de capture et ainsi récupérer le login/pass de l'utilisateur. La conséquence est immédiate pour ce dernier.

Une première tentative serait de crypter le réseau wifi. Cependant si tout le monde utilise la même clé (cryptage wep ou wpa) ou authentification PSK (pre-shared key), les trames réseaux seront déchiffrables par tout le monde, on se retrouve donc dans la situation hub précédente avec l'accès aux données échangées par tous. Une solution avec authentification 802.1x (certificat ou login/pass) permet de garantir ce point là.En effet 802.1x permet de générer une clé de chiffrement pour chaque utilisateur connecté au point d'accès. Ainsi un utilisateur ne peut déchiffrer la communication d'un autre utilisateur. Il serait intéressant d'apporter cette solution dans notre portail captif, mais ce chiffrement ne peut être géré que par le point d'accès sans-fil ce qui se retrouve hors du cadre de notre projet.

IV Comparaison entre portails captifs

IV.1 Caractéristiques principales



NoCatSplash

- Compatible Radius, LDAP, MySQL
- Ecrit en Perl et exige plusieurs modules.
- Communique avec la passerelle pour informer si l'utilisateur peut passer. Peut régler le débit et spécifier des règles de pare-feu pour un utilisateur.
- Sécurisation du transfert entre le système authentification et la passerelle par clé pgp. La documentation conseille de séparer le serveur d'authentification (droits limités) de la passerelle (droit root pour pouvoir modifier les règles iptables).
- Fenêtre de saisie du mot de passe en https.
- Simplicité au niveau utilisateur, performance
- A besoin d'un navigateur avec pop-up, donc inutilisable sur un PDA.

WifiDog

- Compatible Radius
- Nécessite Apache, php, PostgreSql, Pear.
- Faible consommation en ressource réseau et faible ressource de la passerelle lui permettant d'être installé sur certains routeurs/points d'accès comme le WRT54G.
- Contrôle iptable pour définir les règles de passage du pare-feu.

- Adapté à une communauté et à une gestion d'un parc conséquent de points d'accès avec un monitoring poussé.
- Vérifie l'activité grâce à un ping. Evite d'avoir une pop-up comme dans noCat.

Chilispot

- Compatible Radius
- Nécessite Apache, Mysql, php
- Perte au niveau de la bp et consommation de ressources système
- Authentification WPA possible

Talweg

- Compatible Radius
- Récupère les demandes d'accès aux pages web et les retourne dans une connexion https
- Limite l'accès aux ports 80 et 443 de part son fonctionnement

IV.2 Choix

Les solutions retenu seront NoCat, car déjà implémenté sur l'université et WifiDog car nous souhaitons examiner la fiabilité du système par ping permettant de savoir si le client est toujours connecté.

V Test et comparatif entre NoCat et WifiDog

V.1 Matériel requis

Afin d'être plus rapide dans notre phase de test entre WifiDog et NoCat, deux serveurs seront utiles. Une carte wifi de type Atheros (pour sa compatibilité sous Linux) sera aussi nécessaire pour mettre en évidence les possibilités d'usurpation d'identités et de récupération des login/mot de passe de l'utilisateur.

V.2 Installation et configuration succinct de NoCat

Le logiciel NoCat est un portail web captif destiné à "sécuriser" le partage d une connexion sans-fil en redirigeant les utilisateurs vers une page web sur laquelle ils doivent s authentifier via https avec un login/mot de passe. Pour cela, NoCat modifie dynamiquement les règles iptables du firewall pour ouvrir certains ports pour l utilisateur (uniquement TCP avec NoCatAuth-0.82). Remarque : NoCatAuth nécessite les droits root pour manipuler les règles iptables.

Le logiciel NoCat fournit 2 modules : authserv (serveur web d authentification) et Gateway (passerelle). La documentation conseille d installer les 2 modules sur 2 machines séparées car d une part la Gateway tourne avec les droits root et d autre part cela permet de centraliser la gestion des droits d accès. Cependant, pour des contraintes matérielles, la passerelle et le serveur d authentification pourront être installés sur la même machine.

Services nécessaires

Un serveur web Apache-ssl devra être mis en place lors de l'installation de la distribution Linux.

Le logiciel NoCat nécessite que les logiciels suivants soient installés :

- iptables
- Perl 5.x.x
- The Perl5 Database Interface (DBI)
- mySQL database interface to Perl *optionnel : on peut stoquer les logins/mots de passe dans un fichiet texte ou dans une base de données MySQL).*
- MD5 Message Digest for Perl
- Public Key encryption system(pgp)

Déroulement de l'installation

- Télécharger le package NoCatAuth sur le site NoCat puis le décompresser.
- Installation de la passerelle (gateway)
- Créer une clé PHP pour les échanges entre la passerelle et le serveur d authentification
- Copier la clé créée du serveur d authentification dans le répertoire PGP de la passerelle
- Le serveur web d authentification a besoin des droits en écriture sur le répertoire PGP. Changer le owner et le group du répertoire pour qu ils soient les mêmes que ceux du serveur web Apache (par défaut www-data sous Debian)
- Configuration du fichier /usr/local/nocat/gateway/nocat.conf
- Configuration du DNS
- Configuration du fichier /usr/local/nocat/authserv/nocat.conf
- Configuration du serveur Apache-SSL et génération du certificat
- Authentification des utilisateurs avec le fichier *passwd* ou avec une base SQL
- Puis test du Hotspot.

V.3 Installation et configuration succinct de WifiDog

WifiDog permet d'avoir un système centralisé de contrôle des accès, un système de répartition de la bande passante et même de diffuser du contenu spécifique à un hotspot donné. Il fonctionne avec n'importe quel navigateur (pas de Javascript, marche aussi avec les PDA). Il est développé en C et spécialement pour le fameux WRT54G, mais il fonctionne aussi ailleurs (sur n'importe quel Linux récent). Une installation classique occupe 30kb en architecture i386.

Le portail est codé en PHP/Postgresql.

Logiciels nécessaires

Apache2, php5 et mysql sont nécessaires pour la mise en place du portail d'authentification wifidog.

Wifidog requiert une base de données PostgreSQL, nous allons donc l'installer avec son support pour php5.

- installer PostgreSQL et le support php5
- installer Phlickr, téléchargeable à l'adresse suivante: http://prdownloads.sourceforge.net/phlickr/Phlickr-0.2.5.tgz
- récupérer la dernière version de wifidog-auth sur le serveur.
- Création de la base de données PostgreSQL et de l'utilisateur
- Ouvrir le navigateur à l'adresse suivante: http://Ip_de_votre_serveur/wifidog-

auth/wifidog/install.php

– Ensuite notez le mot de passe contenu dans le fichier /tmp/dog_cookie.txt créé précédemment pour pouvoir se connecter sur la page Web d'installation de WifiDog. On pourra alors installer les modules manquant et configurer le serveur.

VI Architecture logiciel

VI.1 Analyse

Nous allons ici analyser le fonctionnement logiciel des portails captifs NoCat et WifiDog. Ces deux logiciels ont beaucoup de points communs.

La partie pare-feu est gérée par iptables. Le principe est d'utiliser la classification du trafic grâce aux tables mangle et filter afin que le trafic correspondant à un certain couple mac/ip soit marqué. Lors de l'initialisation du portail captif, celui-ci rajoute des règles au pare-feu permettant au trafic marqué d'accéder au réseau. Il est aussi possible, grâce à la classification du trafic, de faire une QoS ou d'allouer des règles selon les utilisateurs authentifiés. Dans le cas de NoCat, c'est un script bash qui se charge d'ajouter ou de supprimer les règles; dans celui de WifiDog, c'est un programme codé en C qui s'en charge.

La partie redirection est en fait un serveur web très basique. Lors de la première tentative de connexion en http de la part du client, la passerelle va le rediriger vers ce serveur d'authentification grâce au code http 302. Ce serveur web est en C pour WifiDog et en Perl pour NoCat.

La partie d'authentification est gérée principalement par un serveur web. Celui-ci va alors lui fournir une page lui proposant de s'identifier. Cette page est cryptée par ssl pour assurer la sureté de la saisie du couple login/mot de passe. Ensuite le serveur peut interroger diverses base de données comme LDAP, PostgreSQL, MySQL ou utiliser un système comme RADIUS. Après la validation de la phase d'authentification, le pare-feu est informé que le client, identifié par son couple mac/ip, est autorisé à traverser la passerelle. Cette dernière étape est effectuée différemment par ces deux portails captifs.

Pour NoCat, le service d'authentification va contacter directement la passerelle en lui envoyant un message crypté par PGP pour lui dire quel couple mac/ip est autorisé à passer.

Pour WifiDog, le service d'authentification redirige le client vers la passerelle avec un mot-clé, appelé token, contenu dans l'url. La passerelle va vérifier la validité de ce token en interrogeant le serveur d'authentification qui va lui indiquer si ce token est valide.

A noter que le couple mac/ip est détecté par le service d'authentification dans le cas de NoCat et par la passerelle (en se basant sur sa table Arp) dans le cas de WifiDog.

Enfin il reste maintenant à gérer la durée de connexion. En effet, l'utilisateur ne va pas forcément penser à cliquer sur le bouton « Logout », il faut donc introduire un système de « Timeout » pour gérer ces cas en scrutant la présence de l'utilisateur sur le réseau.

Concernant NoCat, celui-ci ouvre une pop-up après la réussite de l'authentification qui va se rafraichir à intervalle régulier afin de signaler à la passerelle que l'utilisateur est toujours actif sur le réseau. Ce système implique donc à l'utilisateur d'avoir un navigateur web gérant les popup, de le garder ouvert et d'exécuter du javascript pour gérer le rafraichissement automatique.

Concernant WifiDog, celui-ci pingue à intervalle régulier le client afin de vérifier qu'il est toujours présent sur le réseau. Le client peut ainsi fermer son navigateur web après avoir passé la phase d'authentification.

VI.2 Bilan

En étant codé en C, WifiDog a fait le choix d'avoir un exécutable le plus petit possible le permettant de l'intégrer à certains points d'accès/passerelle comme le WRT54 limité en ressource processeur et mémoire. Néanmoins le développement en C est plus long et nécessite de la part du programmeur une plus grande attention.

NoCat, codé en Perl, nécessite l'installation de l'interpréteur Perl ainsi que divers modules pour pouvoir communiquer avec les base de données. Cependant ce langage permet un développement plus rapide et l'interpréteur Perl permet d'éviter divers problèmes lors de l'exécution du code.

L'ajout de fonctionnalités au niveau des règles de filtrage réseaux se fera à l'aide d'iptables.

VII Cahier des charges

- Comparaison approfondie entre WifiDog et NoCat.

Nous souhaitons étudier WifiDog car sa fenêtre de capture codée en Php permet de la modifier facilement pour l'adapter à nos besoins et d'y rajouter des fonctionnalités côté utilisateur. De plus, nous voulons voir la fiabilité du système par ping permettant de savoir si le client est toujours connecté.
Les critères suivants seront examinés durant cette étude:

  - Installation, configuration et administration du portail captif.
  - Confort au niveau utilisateur: compatibilité au niveau des navigateurs, rapidité de la phase d'authentification, changement de points d'accès sans devoir se ré-authentifier.
  - Performance: Consommation de ressources au niveau de la passerelle, faible impact sur la bande passante du wifi et de la latence de la connexion.
  - Sécurité: usurpation de la passerelle dans le but d'obtenir le login/mots de passe de l'utilisateur, utilisation de la connexion d'un utilisateur authentifié.

- Fonctionnalités à implémenter du côté client:
  - Rendre la page d'accueil du portail captif plus attrayante et lui rajouter des pages comme un mode d'emploi, des informations sur l'utilisateur connecté, une liste des points d'accès (sans fil et filaire) de l'université.
  - Permettre une connexion sécurisée au réseau de l'université (802.1x ou VPN grâce à un logiciel dédié).
- Fonctionnalités à implémenter au niveau serveur:
  - Permettre une gestion des utilisateurs plus fines: accès à certains vlans, bande passante privilégiée pour certaines utilisateurs, règles de filtrage spécifiques.
  - Détecter des clients « pirates » tentant de tromper les utilisateurs.

VIII Planning prévisionnel

WBS	Name	Start	Finish	Work	Duration	Slack	Cost	Assigned to
1	**Récuparation du matériel et installation**	**jan 21**	**jan 22**	**4d**	**2d**	**38d**	**0**	Olivier
1.1	Installation NoCat	jan 21	jan 22	2d	2d	38d	0	Mathieu
1.2	Installation WifiDog	jan 21	jan 22	2d	2d	38d	0	Thomas
2	Début test hotspot	jan 23	fév 4	1d	10d	39d 1h	0	Olivier, Thomas
3	Install Radius ldap	jan 23	fév 4	1d	10d	39d 1h	0	Mathieu
4	Implémentation du ping	fév 4	mar 6	1d	20d	42d 5h	0	Thomas
5	Implémentation de la QOS	fév 4	mar 6	1d	20d	42d 5h	0	Mathieu, Olivier
6	Implémentation du VPN	mar 6	avr 3	20d	20d	42d 5h	0	Mathieu
7	Détection d'intrusions	mar 6	mar 20	20d	10d	52d 5h	0	Olivier, Thomas
8	compte rendu, conclusion	mai 26	jun 3	7d	7d		0	Mathieu, Olivier, Thomas